



Research article

Mathematical modeling of a Hybrid Mutated Tunicate Swarm Algorithm for Feature Selection and Global Optimization

Turki Althaqafi*

Computer Science Department, School of Engineering, Computing and Design, Dar Al-Hekma University, Jeddah, Saudi Arabia

* **Correspondence:** Email: tthaqafi@dah.edu.sa.

Abstract: The latest advances in engineering, science, and technology have contributed to an enormous generation of datasets. This vast dataset contains irrelevant, redundant, and noisy features that adversely impact classification performance in data mining and machine learning (ML) techniques. Feature selection (FS) is a preprocessing stage to minimize the data dimensionality by choosing the most prominent feature while improving the classification performance. Since the size data produced are often extensive in dimension, this enhances the complexity of search space, where the maximal number of potential solutions is 2^n for n feature datasets. As n becomes large, it becomes computationally impossible to compute the feature. Therefore, there is a need for effective FS techniques for large-scale problems of classification. Many metaheuristic approaches were utilized for FS to resolve the challenges of heuristic-based approaches. Recently, the swarm algorithm has been suggested and demonstrated to perform effectively for FS tasks. Therefore, I developed a Hybrid Mutated Tunicate Swarm Algorithm for FS and Global Optimization (HMTSA-FSGO) technique. The proposed HMTSA-FSGO model mainly aims to eradicate unwanted features and choose the relevant ones that highly impact the classifier results. In the HMTSA-FSGO model, the HMTSA is derived by integrating the standard TSA with two concepts: A dynamic s-best mutation operator for an optimal trade-off between exploration and exploitation and a directional mutation rule for enhanced search space exploration. The HMTSA-FSGO model also includes a bidirectional long short-term memory (BiLSTM) classifier to examine the impact of the FS process. The rat swarm optimizer (RSO) model can choose the hyperparameters to boost the BiLSTM network performance. The simulation analysis of the HMTSA-FSGO technique is tested using a series of experiments. The investigational validation of the HMTSA-FSGO technique showed a superior outcome of 93.01%, 97.39%, 61.59%, 99.15%, and 67.81% over diverse datasets.

Keywords: Feature Selection; Tunicate Swarm Algorithm; machine learning; Rat Swarm Optimizer; classification

Mathematics Subject Classification: 94A60, 11T71, 68P25

1. Introduction

FS is nothing but an ML model, which is utilized to decrease the dimension of a data set [1]. Its main intention is to pick the significant feature for an analytical method, allowing the ones that are terminated or will not deliver beneficial data, which contain numerous advantages. One vital advantage is enhanced interpretability, which will be mainly helpful if the model decisions want to be clarified to human consumers [2]. Furthermore, FS can strengthen performance and simplification by eliminating noise from the data and decreasing the number of features that reduce the danger of overfitting the training dataset. Last, FS can result in forecast times and quicker training owing to the diminished dimension of the dataset [3]. Generally, numerous techniques are accessible and can be acquired to achieve FS. One of the best choices is using filter models that trust the numerical events of the feature to recognize the appropriate one. These models are independent of ML techniques and are used in any method [4]. Otherwise, wrapper models can be employed, which includes training the method with dissimilar mixtures of features and choosing the grouping that produces the finest identification performance. Whereas many are more mathematically costly than filter models, wrapper techniques consider the contact between features and method, foremost to more excellent performance [5].

Embedded models constructed using ML techniques can be employed for FS by classifying the related features over a mixture of FS and model training. Hybrid models that merge the powers of dissimilar techniques are chosen for FS, such as utilizing a filter model to pre-pick features and a wrapper model to perfect the choice [6]. In FS, addressing the optimum sub-set is a vital problem. Complete exploration can yield every possible sub-set by inspecting the full set of features [7]. This method is impossible for a vast dataset and has a very high computing cost since if a dataset grasps M features, then 2^M sub-sets of features are present [8]. In the preceding dual years, meta-heuristics have established their efficacy and productivity in resolving the challenging and larger-scale issues in ML, data mining, and engineering design applications. The mid-level models are proposed to develop further representation to aid high-level statistical techniques [9]. There are three basic classes of these techniques: Evolutionary-based (for instance, evolutionary and genetic models), physics-based, and swarm-based (for instance, ant and bee colonies). While utilizing these techniques, dual different criteria are accessible, such as intensification (exploitation of the finest solution obtained) and diversification (exploration of the searching space) [10]. As indicated, swarm models derive the search mechanism of animals such as moths, bats, ants, cuckoos, etc. Currently, swarm techniques are projected and have shown a valued performance for numerous FS tasks.

I am motivated by the illustrated efficiency of meta-heuristic approaches, specifically swarm-based models, in addressing convolutional and large-scale threats across ML, engineering design applications, and data mining. These methods, inspired by natural behaviors seen in animals such as bats, moths, and ants, present robust outcomes for optimization tasks by balancing intensification (exploitation of the optimum outcome) and diversification (exploration of the search space). I also aims to advance these techniques by presenting the hybrid model, incorporating novel mutation scenarios to improve global optimization procedures and feature selection. The method contributes

substantially to enhancing the efficiency and effectualness of solving real-world issues where precise feature selection and optimization are significant, thereby improving the field's abilities to handle growing convolutional data-driven threats.

I develop a Hybrid Mutated Tunicate Swarm Algorithm for FS and Global Optimization (HMTSA-FSGO) technique. The proposed HMTSA-FSGO model mainly aims to eradicate unwanted features and choose the relevant ones that highly impact the classifier results. The HMTSA-FSGO model, the HMTSA is derived by integrating the standard TSA with two concepts: Dynamic s-best mutation operator for optimal trade-off between exploration and exploitation and directional mutation rule for enhanced search space exploration. The HMTSA-FSGO model also includes a bidirectional long short-term memory (BiLSTM) classifier to examine the impact of the FS process. The hyperparameters can be chosen by rat swarm optimizer (RSO) to boost the Bi-LSTM model performance. The experimental values highlighted that the HMTSA-FSGO model gains better performance over other models. The novel contributions of the HMTSA-FSGO model are as follows:

- The HMTSA-FSGO technique introduces the HMTSA model, which innovatively incorporates the standard TSA with dynamic s-best mutation and directional mutation rules. This combination improves the models' ability to balance exploration and exploitation during optimization tasks, enhancing their efficiency in exploring convolutional search spaces. The model optimizes parameters, which is ideal for robust performance in dynamic environments, namely plant disease classification with BiLSTM.
- The HMTSA-FSGO model integrates the BiLSTM classifier to evaluate the impact of FS, joining its sequential learning capabilities to improve the accuracy of the disease classification. This incorporation explores the efficiency of FS to optimize data representation for enhanced detection results in agricultural contexts
- The presented technique also utilizes the RSO model for optimizing hyperparameters, improving the accomplishment of the BiLSTM network. This confirms an optimum setting custom-made for accurate and efficient recognition of plant disease, enhancing the accuracy of the detection in agricultural applications
- The novelty of the HMTSA-FSGO method is in the incorporation of HMTSA with BiLSTM techniques for plant disease classification, presenting a fusion model that integrates evolutionary swarm intelligence with DL approaches to attain improved robustness and accuracy in disease detection

2. Literature review

Houssein et al. [11] introduced an innovative variation of the Coati Optimizer Algorithm (COA), named eCOA. The developed eCOA includes the RUNge Kutta Optimizer (RUN) and COA techniques. The Enhanced Solution Quality (ESQ) and Scale Factor (SF) device from RUN were used to solve the higher faults of COA. Also, the eCOA is very useful for dual and multi-class identification of feelings utilizing multi-layer perceptron neural networks (MLPNNs). In [12], a hybrid ML technique incorporating dual FS models and a Bayesian optimizer (BO) technique is projected. The method employed dual Random Forest (RF) models of feature significance. Depending on the FS outcomes, ten techniques were proposed and equated: (1) 5 separate ML methods with RF, Bagged Trees Regression (BTR), Classification and Regression Trees (CART), SVR, and MLP; and (2) similar methods adjusted by the BO models. Mostafa et al. [13] projected a novel modified Gorilla group

optimizer (mGTO) technique using a set of operators. The combination of CICD and TFO was used to optimize the exploitation capability. Kwakye et al. [14] proposed a hybrid SI-based MA named Particle Swarm-guided Bald Eagle Search (PS-BES) model. The technique uses the rapidity of PS to guide BE in certifying an even changeover of the technique from exploration to exploitation. Furthermore, the method presents the Attack-Retreat-Surrender model, a novel local-optimal escape approach to improve the balance among intensification and diversification of PS-BES.

Houssein et al. [15] projected a wrapper FS technique that unites the RSO model to evade local optimum. In the projected technique, the transfer function (TF) is inserted to balance global and local search by adapting a constant searching space into a separate space. In [16], a new hybrid metaheuristic technique named SSA-FGWO is proposed. The hybrid tool contains dual stages, such as solid exploitation, to upgrade the leaders' location in the chain populace. In [17], an advanced technique, namely GNDAOA, employed three components: Generalized Normal Distribution Optimizer (GNF), Arithmetic Optimizer Algorithm (AOA), and OBL. Xu et al. [18] projected a CPA variant called the Covariance Gaussian Cuckoo Colony Predation Algorithm (CGCPA) model. The intended GC variable approach was mainly employed to reorganize the agent populace in CPA. In [19], two improved TSA models are introduced, OCSTA and COCSTA, integrating chaos theory, opposition-based learning (OBL), and Cauchy mutation. OCSTA employs static and dynamic OBL, while COCSTA implements centroid opposition-based computing. In [20], the Multi-Strategy Hybrid Harris Hawks Tunicate Swarm Optimization Algorithm (MSHHOTSA) model is introduced. This method integrates hyperbolic tangent domain modification and utilizes a non-linear convergence factor. The HHO model is also used.

Alizadeh et al. [21] present a model by integrating meta-heuristic models and various mechanisms. It also uses salp swarm optimization (SSO) and African vulture optimization algorithm (AVOA) methods for optimization. Furthermore, opposition-based learning (OBL) and β -hill climbing (BHC) approaches are incorporated with the AVOA-SSA model. Pan, Lei, and Wang [22] propose the Discrete Extended Permutation Flow Shop Scheduling Problem (DEPMSP) model by incorporating factory and machine assignments into an extended machine assignment model, utilizing a Knowledge-Based Two-Population Optimization (KTPO) method to minimize energy utilization and delay concurrently. Zhao, Di, and Wang [23] focused on the energy-efficient Distributed Blocking Flow Shop Scheduling Problem (EEDBFSP) employing a hyperheuristic with Q-learning (HHQL), which chooses low-level heuristics (LLHs) based on historical feedback to minimize total tardiness (TTD) and total energy consumption (TEC). Zhao et al. [24] present an Improved Iterative Greedy (IIG) model, employing the Variable Neighborhood Descent (VND) method with perturbation operators based on critical factories. It also integrates a Q-learning mechanism to select weighting coefficients.

Limitations and research gaps

The existing studies need to be revised in various areas. For instance, eCOA integrates RUN with COA, potentially enhancing computational complexity. Hybrid ML models with Bayesian optimization may need help with high-dimensional spaces and scalability. The mGTO and PS-BES methods must balance algorithm parameters for efficient synergy, while RSO-based wrapper FS needs careful parameter tuning for robustness. SSA-FGWO, GNDAOA, OCSTA, COCSTA, and MSHHOTSA each face limitations in optimizing various and sometimes conflicting strategies. Incorporating SSO, AVOA, OBL, and BHC needs cautious coordination and parameter management

to reduce computational complexities and ensure efficient accomplishment across varied optimization tasks. A notable research gap is in the comparative analysis of the HMTSA against other recent meta-heuristic optimization models in terms of feature selection efficiency and computational accomplishment. Moreover, more exploration needs to be conducted into the specific impact of HMTSA's incorporated mutation strategies to optimize complex search spaces, specifically within the context of agricultural data and BiLSTM-based disease classification methods. Addressing these gaps could provide valuable insights into the effectiveness and applicability of the models across diverse optimization scenarios.

3. Methodology

In this work, an HMTSA-FSGO method is presented. The proposed HMTSA-FSGO model mainly aims to eradicate the redundant features and choose the relevant ones that highly impact the classifier results. Figure 1 represents the working flow of the HMTSA-FSGO method.

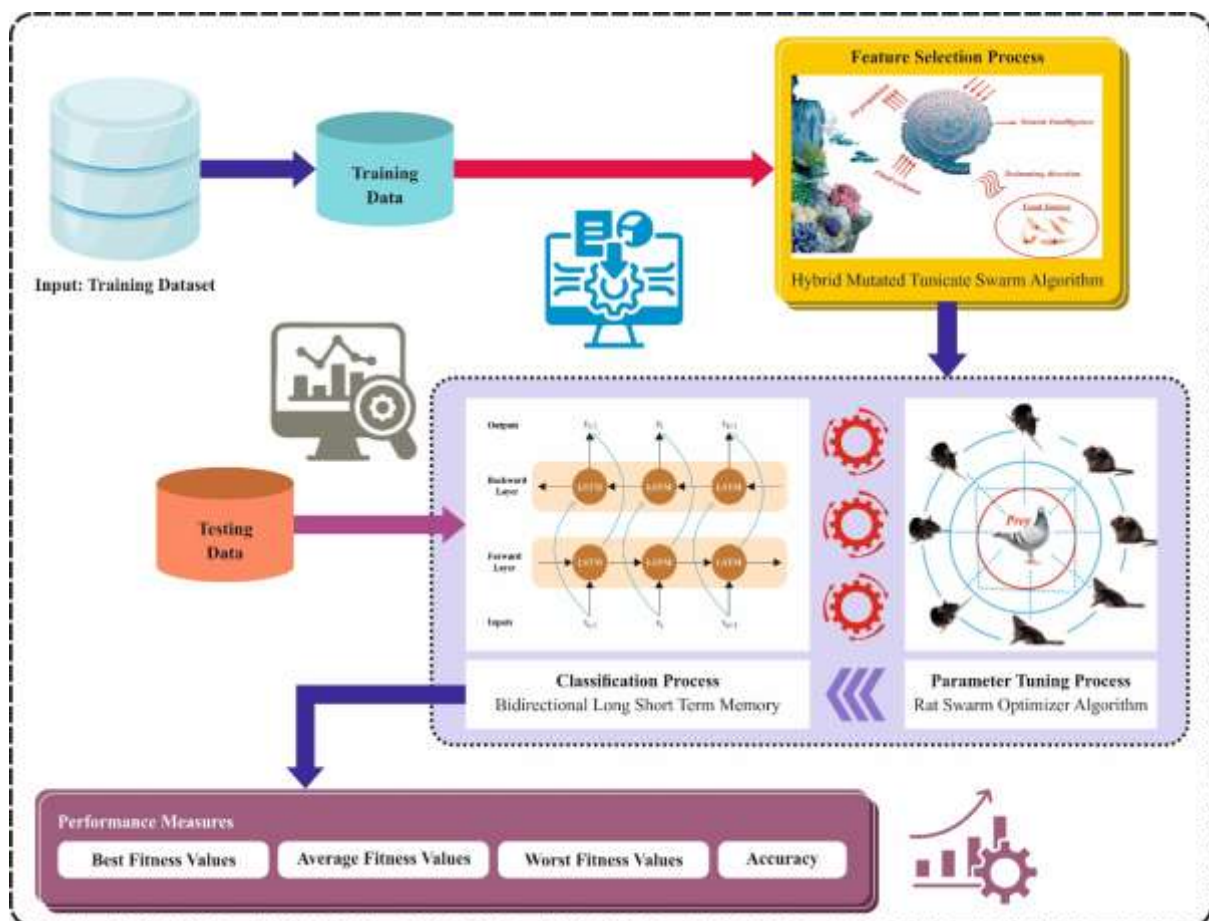


Figure 1. Overall flow of HMTSA-FSGO technique.

3.1. Process involved in HMTSA

In 2024, Chandran and Mohapatra proposed a new metaheuristic approach named the TSA that

stimulates the foraging behaviors of tunicates [25]. All the tunicates are cylindrical and show a gelatinous tunic that helps to connect each tunicate. However, TSA is based on two dissimilar behavioral patterns of tunicate in the deep sea, such as swarm intelligence and jet propulsion to discover food sources (viz., the optimum solution). Figure 2 illustrates the flowchart of the TSO model, and also the mathematical modelling of TSA is given below:

- Avoiding conflict between the search individual.
- Moving to the optimum search individuals.
- Converge towards the area surrounding the optimum search individuals.

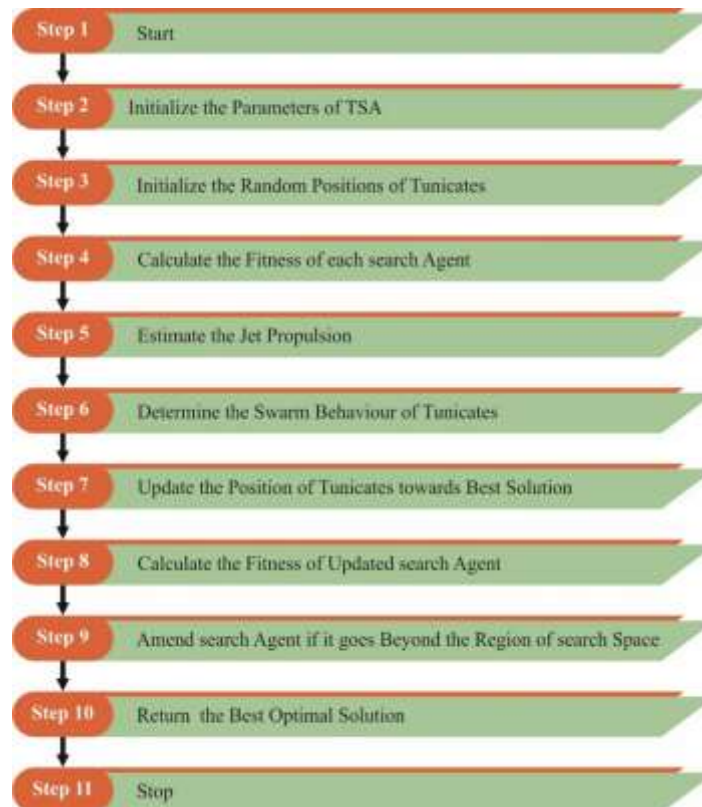


Figure 2. Flowchart of TSA.

On the other hand, the swarm intelligence technique helps to update the tunicate's location according to the optimum solution.

Avoiding conflict between the search individuals: The \vec{A} vector determines the updated location of the individual to avoid conflict between the search individuals as follows:

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \quad (1)$$

$$\vec{G} = r_2 + r_3 - \vec{F} \quad (2)$$

$$\vec{F} = 2 * r_1 \quad (3)$$

$$\vec{M} = [P_{min} + r_1 \cdot (P_{max} - P_{min})] \quad (4)$$

Here, \vec{M} shows the social forces between the tunicates. The gravitational force and the water stream rate from the deep sea are \vec{G} and \vec{F} , respectively. r_1 , r_2 , and r_3 are uniform distribution random numbers in $[0,1]$. P_{min} and P_{max} are the initial and subordinate speeds of tunicates fixed as 1 and 4, respectively.

Moving to the better search, individual: After avoiding the conflict between the tunicates, everyone proceeded to the optimum tunicates. Eq (5) determines the mathematical formula for approaching the optimum tunicates.

$$\vec{SD} = |F_{best} - rand * X(t)| \quad (5)$$

where F_{best} is the food location, $X(t)$ denotes the tunicate location, and $rand \in [0,1]$, \vec{SD} represents the spatial distance from the tunicates to the food source.

Converging to the region nearby the better search for individuals: The tunicate converges to the location of optimum tunicates as denoted in Eq (6) and Eq (7).

$$X(t) = F_{best} + \vec{A} \cdot \vec{SD}, \text{ if } rand \geq 0.5 \quad (6)$$

$$X(t) = F_{best} - \vec{A} \cdot \vec{SD}, \text{ if } rand < 0.5 \quad (7)$$

Now, $X(t)$ shows the updated location of the tunicate concerning the food location F_{best} .

Swarming behavior of tunicates: In the swarm intelligence, the tunicate position is updated according to the locations of the first two optimum tunicates.

$$X_i(t+1) = \begin{cases} \frac{X_i(t) + X_{i-1}(t+1)}{2 + r_1}, & \text{if } i > 1 \\ X_i(t), & \text{if } i = 1 \end{cases} \quad (8)$$

In Eq (8), $i = 1, 2, \dots, N$, N indicates the population size, $X_i(t+1)$, and $X_{i-1}(t+1)$ are the updated locations of the present and prior search individuals at the next iteration.

Adaptive s-best mutation scheme: One of the primary key features is the exploration-exploitation balance used to search capability better [26]. In the HMTSA technique, the present solution generates a new solution with no bias, and the continuing shift from exploration to exploitation is lost. Now, the S -optimum mutation is presented. The s -optimum mutation randomly chooses one from the S range to generate a new solution. Here, the i -th updated solution is produced by mutating any solution arbitrarily selected from the topmost S or mutating the present solution. Parameter S is nonlinearly reduced, which enables the choice of a different search space; however, the solution selected becomes more constrained. This improves the diversity and good search capability while it shifts to exploitation. During the exploitation, it focuses on a smaller range of selective solutions near the global optimum.

$$y = X_{sbest} + F \times (X_{r1} - X_{r2}) \quad (9)$$

where X_{sbest} is the solution from the top $S \times N$ solutions rather than the deterministic optimum solution, the X_{r1} and X_{r2} are arbitrarily chosen entities from the whole generation, which satisfy

that $r1$ is not equivalent to $r2$. Either is comparable to the existing individual. F are a constant chosen within $[0,1]$, and S is linearly dropped.

$$S_t = 1 - \left(1 - \frac{1}{N}\right) \times \frac{t - 1}{T - 1} \quad (10)$$

In Eq (10), t refers to the existing time, T denotes the maximum iteration count, and N indicates the population size. Hence, the significant value of S occurs in the solution for better exploration, whereas in the later generation, S becomes small to function on the exploitation enhancement.

Directional mutation rule: Modification of the searching agent's direction makes it possible to find within the search range efficiently. The mutated solutions exploit the guiding data without any bias to any direction, and the d -directed values are added to Eq (9).

$$y = X_{sbest} + d \times F \times (X_{r2} - X_{r3}) \quad (11)$$

d value is altered with the fitness of the arbitrarily chosen solution as follows.

$$d = \begin{cases} 1 & \text{if } f_{X_{r2}} < f_{X_{r3}} \\ -1 & \text{otherwise} \end{cases} \quad (12)$$

Nearest neighbor comparison (NNC): The concept trial gets an assessment value worse than the present solution. This mainly occurs in a metaheuristic algorithm, which causes wasted time, and the search arrow moves farther from the global optimum solution. NNC is integrated into the adaptive mutation approach and added with the HMTSA technique to prevent and decrease redundant function assessments. NNC exploits the k -nearest neighbour (kNN) predictor to judge whether the newest solution is worth assessing. Bi-LSTM predictor relies on active learning that decreases the strict assessment number via the search techniques. For trial vector v , nearby neighbour X_n is discovered through the distance measure represented in Eq (13).

$$d(x, v) = \sqrt{\sum_{i=1}^D \left(\frac{X_i - v_i}{x_{\max} - X_{\min}}\right)^2} \quad (13)$$

$d(x, v)$ denotes the distance between x and v vectors. x_{\max} , X_{\min} are the maximal and minimal values of vector X . So, X_n refers to the minimum distance to the vector v .

Compare X_n with the solution. If X_n worsens, the trial vector skips it; otherwise, it will assess.

3.1.1. Application of Swarm Intelligence Model in FS

Swarm intelligence optimization models are extensively utilized in FS to improve the effectiveness and efficiency of ML techniques. These approaches, motivated by the overall characteristics of natural swarms, namely bees, birds, and ants, utilize populace-based search strategies to navigate convolutional feature spaces. They also outperform in balancing exploration (searching for diverse subsets of features) and exploitation (refining the best feature subsets) for identifying optimum feature combinations that optimize anticipative accomplishment while minimizing computational costs. By iterative computing and choosing subsets of features depending on fitness criteria, swarm

intelligence algorithms, namely Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithm (GA), contribute crucially to automating the FS procedure, enhancing the interpretability of the method, and improving the comprehensive predictive accuracy in several areas comprising image processing, financial forecasting, and bioinformatics [27].

3.1.2. Fitness function of HMTSA

The feature combination typically has 2^N in FS if the feature vector size is N [28]. The hybrid mechanism dynamically explores the search range and generates a better feature combination. FS falls within a multiobjective problem since it should fulfil multiple objectives to get a better solution that increases the performance and decreases the selected features. The fitness function to determine a solution is constructed to balance both objectives.

$$fitness = \alpha \Delta_R(D) + \beta \frac{|Y|}{|T|} \quad (14)$$

In Eq (14), $\Delta_R(D)$ is the classification error rate. $|Y|$ refers to the size of the subset, and $|T|$ shows the overall number of features. α refers to parameter $\in [0, 1]$ concerning the weight of the classifier error rate, correspondingly and $\beta = 1 - \alpha$ specifies the importance of the feature reduction. The effects will be the neglect of solutions if the assessment function considers the classifier accuracy, which might have similar accuracy and lesser features chosen that serve as a critical factor in decreasing the problem's dimensionality.

3.2. Classifier selection

Next, the HMTSA-FSGO model also includes a BiLSTM classifier to examine the impact of the FS process. LSTM is a special RNN with a recurrent hidden layer (HL) named memory units [29].

3.2.1. BiLSTM Model

Input Sequences: Input data, generally sequences of feature vectors or tokens, are given using the BiLSTM method. Each input sequence can portray a time series, text sequence, or sequential data.

Bidirectional Processing: The BiLSTM technique comprises two LSTM networks: one processes the input sequence forward in time (from the beginning to the end), and the other processes it backwards (from the end to the beginning). This bidirectional processing assists in capturing dependencies and patterns in both directions of the input sequence.

Hidden State Calculation: As the input sequence is processed via every LSTM unit (forward and backwards), hidden states are computed at every time step. These hidden states encapsulate data about the input sequence up to that point, considering both past and future contexts due to bidirectional processing.

Concatenation: The outputs (hidden states) from the forward and backward LSTMs at each time step are concatenated. This integrated representation captures an overall understanding of the input sequence related to unidirectional LSTM methods.

Output Layer: The concatenated hidden states are then sent to an output layer, which usually comprises additional processing (such as dense layers or softmax activation) relying on the specific task, namely classification or regression.

Training and Optimization: The entire BiLSTM method is trained using labelled data with appropriate loss functions (e.g., categorical cross-entropy for classification). During training, the model's parameters (weights and biases) are optimized by employing backpropagation through time (BPTT) to minimize the loss function and improve prediction accuracy.

Prediction: Once trained, the BiLSTM model can be used to make predictions on new input sequences. The model's bidirectional architecture and learned representations enable it to capture complex dependencies and patterns in sequential data effectively. It benefits natural language processing (NLP), time series, and biological sequence analysis.

3.2.2. Classification process

Self-connection exists within the memory units. It has forgotten the input and output gates for all the memory cells. The HL of LSTM is termed the LSTM cells. LSTM can plot long-term dependency by determining all the memory cells with gates \mathfrak{R}^d , where the memory size of HL of LSTM is d . The LSTM cells have the input x_t and the output h_t layers at iteration t . LSTM has three gates, namely existing input x_t and HL b_{t-1} : input gate i_t , forget gate f_t , and output gate o_t . Also, it considers the input cell layer s_t , the output cell layer c_t , and the prior output cell layer c_{t-1} while updating and training parameters.

These gates are evaluated as follows.

$$i_t = \sigma(w_{i_x}x_t + w_{i_h}h_{t-1} + b_i) \quad (15)$$

$$f_t = \sigma(w_{f_x}x_t + w_{f_h}h_{t-1} + b_f) \quad (16)$$

$$o_t = \sigma(w_{o_x}x_t + w_{o_h}h_{t-1} + b_o) \quad (17)$$

$$s_t = \tanh(w_{s_x}x_t + w_{s_h}h_{t-1} + b_s) \quad (18)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot s_t \quad (19)$$

$$h_t = \tanh(c_t) \odot o_t \quad (20)$$

where \odot refers to the component-wise product; w_i , w_f , w_o , and w_s denote the weighing factor utilized to map the HL and inputted to the gates mentioned above, and the input cell states; b_i , b_f , b_o , and b_s are bias vectors; and σ and \tanh are the sigmoid and tangent functions.

The last output of LSTM is a vector of output, as stated in Eq (21).

$$Y_t = [h_{t-n}, \dots, \dots, h_{t-1}] \quad (21)$$

The Bi-LSTM links two HLs into one output layer. The backward-layer output series, \vec{h} , is evaluated by the reversed input from time $t - n$ to $t - 1$. The forward layer output series \vec{h} is repeatedly computed by the input in the positive series from time $t - n$ to $t - 1$. Both layers' outputs are evaluated using the typical LSTM updating equation. This representation is concatenated together through the attention module. The BiLSTM layer produces the output vector where all the elements are evaluated in Eq (22).

$$y_t = \sigma(h^{\rightarrow}t, h^{\leftarrow}t) \quad (22)$$

The model could capture several salient words where σ indicates the soft attention function for combining both output sequences. Therefore, the last output of the Bi-LSTM layer is given in Eq. (23).

$$Y_t = [y_{t-n}, \dots, y_{t-1}] \quad (23)$$

3.3. Hyperparameter optimization

To enhance the performance of the BiLSTM model, the RSO technique is implemented in the hyperparameter tuning process [30]. Diverse in size and weight, Rats comprise black and brown rat variants, with females named as does and males as bucks. Rats are well known for their intelligence; they engage in activities like training, chasing, boxing, and jumping. Despite their intelligence, rat aggressiveness may pave the way to the death of some animals. This study focuses on mathematically modelling rat hunting and chasing behaviors to develop the RSO model for optimization objectives.

The proposed RSO technique and the fighting and chasing behaviors of rats are discussed in the following section.

3.3.1. Chasing the prey

Rats are naturally sociable creatures that hunt the target in packs through social behaviors. The optimum searching agent should know the prey location to describe these behaviors. Based on the optimum search agent, the other search agents could update the positions, and it is given below:

$$\vec{P} = A \cdot \vec{P}_i(x) + C \cdot (\vec{P}_r(x) - \vec{P}_i(x)) \quad (24)$$

In Eq (24), \vec{P} is the resulting vector or a position in the searching space, $\vec{P}_i(x)$ implies the current position or solution vector influenced by the parameter x , and $\vec{P}_r(x)$ denotes a reference position or vector also influenced by x , potentially showing a global best solution or a guiding reference point. A and C are scalar coefficients. A scales the influence of $\vec{P}_i(x)$ directly, and C scales the difference between $\vec{P}_i(x)$ and $\vec{P}_r(x)$, adjusting the influence of the reference position relative to the present position.

$$A = R - x \times \left(\frac{R}{\text{Max}_{\text{Iteration}}} \right), \quad (25)$$

where $x = 0, 1, 2, \dots, \text{Max}_{\text{Iteration}}$, and:

$$C = 2 \cdot \text{rand}(\bullet) \quad (26)$$

R depicts a constant or a predefined value related to the optimization procedure. x illustrates the current iteration or some form of iteration index. $\text{Max}_{\text{Iteration}}$ refers to the maximum number of iterations set for the optimization procedure. Also, $\text{rand}(\cdot)$ shows a function that generates random numbers, and 2 represents a constant multiplier. R and C are independent random parameters within [1,5] and [0,2]. The A and C parameters result in the best exploitation and exploration at the iteration.

3.3.2. Fighting with prey

Eq (27) describes the interaction of rats with the target,

$$\vec{P}_i(x + 1) = |\vec{P}_r(x) - \vec{P}|, \quad (27)$$

In Eq (27), $\vec{P}_i(x + 1)$ is the revised following location of the rat. It saves the better solution, changes other solutions' positions, and compares search agents based on the best one. The parameter is changed to attain various locations corresponding to the existing location. Also, this concept is extended in an n -dimensional setting. Thus, the revised values of the A and C parameters ensure exploitation and exploration. Figure 3 defines the flowchart of RSO.

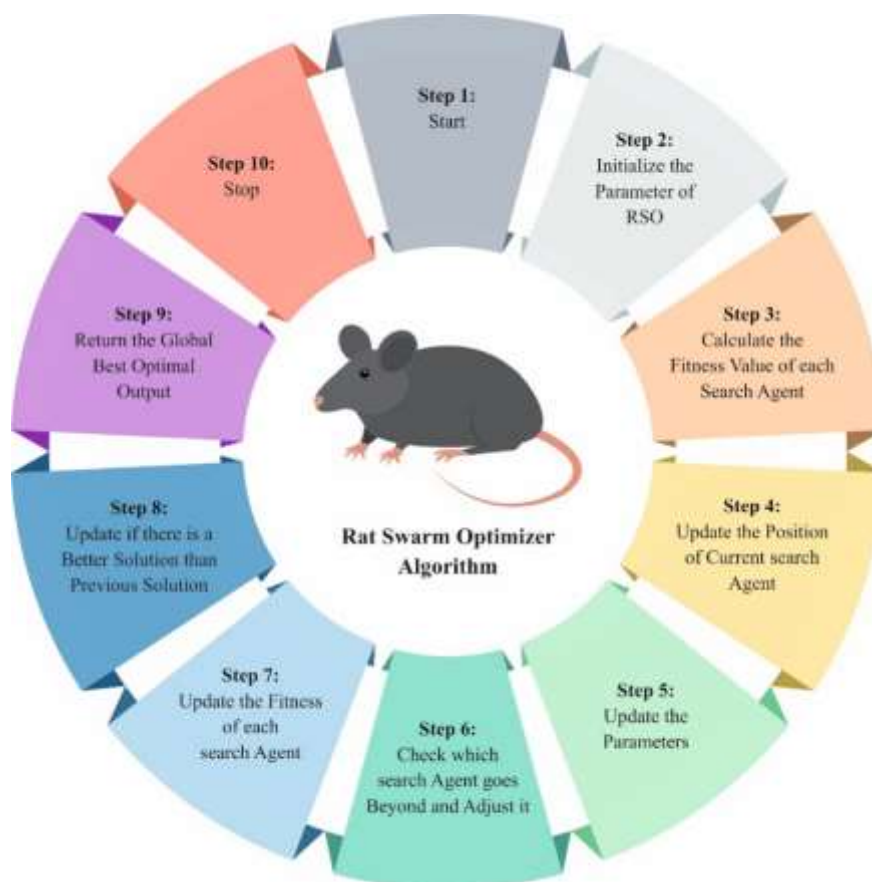


Figure 3. Flowchart of RSO.

The RSO technique derives an FF to attain better classifier results. It describes a positive integer to characterize the superior outcome of the solution candidate. Here, the reduction of the classifier error rate is regarded as the FF.

$$\begin{aligned} fitness(x_i) &= ClassifierErrorRate(x_i) \\ &= \frac{No. of misclassified samples}{Total No. of samples} \times 100 \end{aligned} \quad (28)$$

4. Result analysis and discussion

4.1. Dataset details

The simulation analysis of the HMTSA-FSGO method is tested under five distinct medical datasets [31]. The datasets include various features and are collected from the UCI repository. The details relevant to the dataset are given in Table 1.

Table 1. Details on database.

S. No	Dataset Name	No. of Features	No. of Samples	Classes
Dataset 1	Hepatitis	19	155	2
Dataset 2	Breast cancer	9	699	2
Dataset 3	Lung Cancer	56	32	3
Dataset 4	Dermatology	34	366	6
Dataset 5	Arrhythmia	279	452	16

4.2. Best cost analysis of HMTSA-FSGO technique

Table 2 represents the features obtained by the HMTSA-FSGO technique with existing FS models on all datasets. The HMTSA-FSGO technique has chosen the optimal number of features on all datasets, selecting 8, 4, 23, 17, and 154 features under datasets 1–5.

Table 2. FS on five datasets.

S. No	Dataset Name	No. of Features	Selected Features
Dataset 1	Hepatitis	19	8
Dataset 2	Breast cancer	9	4
Dataset 3	Lung Cancer	56	23
Dataset 4	Dermatology	34	17
Dataset 5	Arrhythmia	279	154

Table 3 and Figure 4 represent the best fitness values (BFVs) of the HMTSA-FSGO technique with other FS models [32]. The results indicate that the ant lion optimization (ALO) and grey wolf optimization (GWO) methods have shown worse BFV values on all datasets. On continuing with the Salp swarm algorithm (SSA), PSO and GA methods have obtained closer BFV values. Although the improved SSA (ISSA) has gained reasonable BFV, the HMTSA-FSGO technique demonstrates optimal BFV values. It is noticed that the HMTSA-FSGO technique accomplishes a lower BFV of 0.0211, whereas the ISSA, SSA, PSO, GA, ALO, and GWO methods obtain higher BFV of 0.0284, 0.0378, 0.0323, 0.0482, 0.0832, and 0.1097, correspondingly.

Table 3. BFV analysis of HMTSA-FSGO technique with other FS methods under five datasets.

Dataset	BFVs						
	HMTSA-FSGO	ISSA	SSA	PSO	GA	ALO	GWO
Hepatitis	0.0560	0.0692	0.0775	0.0861	0.0868	0.1298	0.1961
Breast cancer	0.0037	0.0095	0.0347	0.0346	0.0226	0.0407	0.0400
Lung cancer	0.0221	0.0302	0.0300	0.0470	0.0319	0.1894	0.2536
Dermatology	0.0009	0.0015	0.0174	0.0206	0.0180	0.0192	0.0227
Arrhythmia	0.0228	0.0317	0.0295	0.0396	0.0817	0.0369	0.0361
Average	0.0211	0.0284	0.0378	0.0323	0.0482	0.0832	0.1097

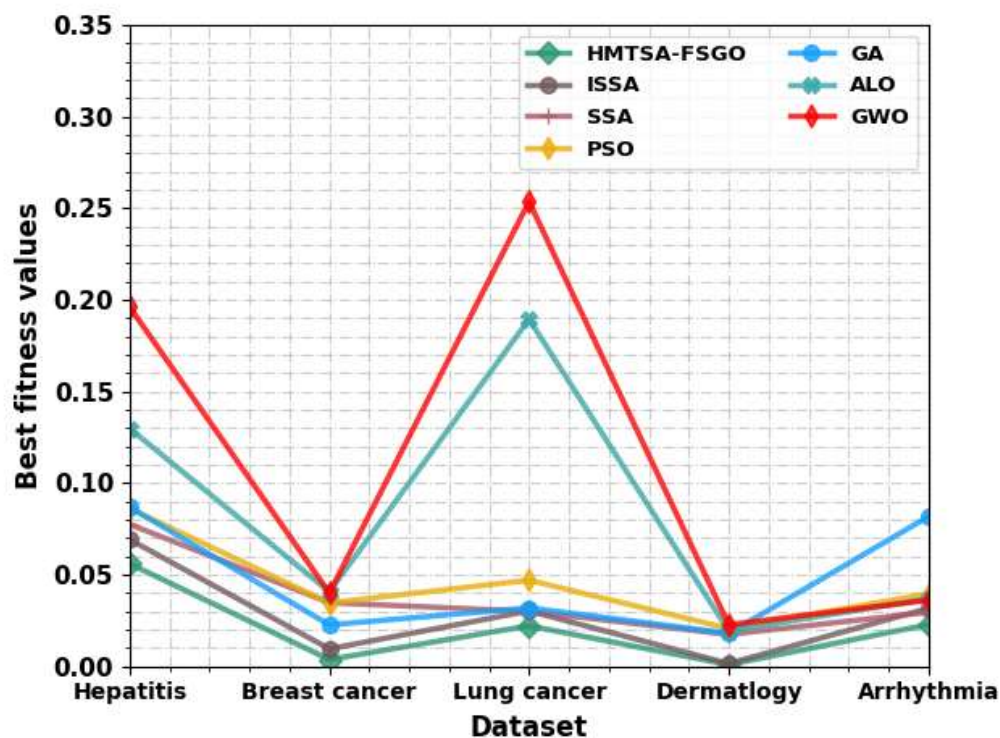
**Figure 4.** BFV analysis of the HMTSA-FSGO technique under five datasets.

Table 4 and Figure 5 characterize the average fitness values (AFVs) of the HMTSA-FSGO technique with other FS methods. The outcomes specify that the ALO and GWO have revealed worse AFV values on all datasets. On continuing with, the SSA, PSO, and GA techniques have gained closer AFV values. Even though the ISSA has attained reasonable AFV, the HMTSA-FSGO method validates optimal AFV values. Note that the HMTSA-FSGO technique achieves a lower AFV of 0.039, whereas the ISSA, SSA, PSO, GA, ALO, and GWO methods obtain higher AFV of 0.054, 0.062, 0.100, 0.099, 0.085, and 0.121, correspondingly.

Table 4. AFV analysis of HMTSA-FSGO technique with other FS models under five datasets.

Dataset	AFVs						
	HMTSA-FSGO	ISSA	SSA	PSO	GA	ALO	GWO
Hepatitis	0.093	0.111	0.121	0.161	0.151	0.130	0.220
Breast cancer	0.011	0.020	0.040	0.039	0.038	0.043	0.042
Lung cancer	0.063	0.091	0.071	0.151	0.173	0.191	0.285
Dermatology	0.003	0.004	0.031	0.091	0.043	0.022	0.023
Arrhythmia	0.026	0.045	0.047	0.061	0.091	0.038	0.037
Average	0.039	0.054	0.062	0.100	0.099	0.085	0.121

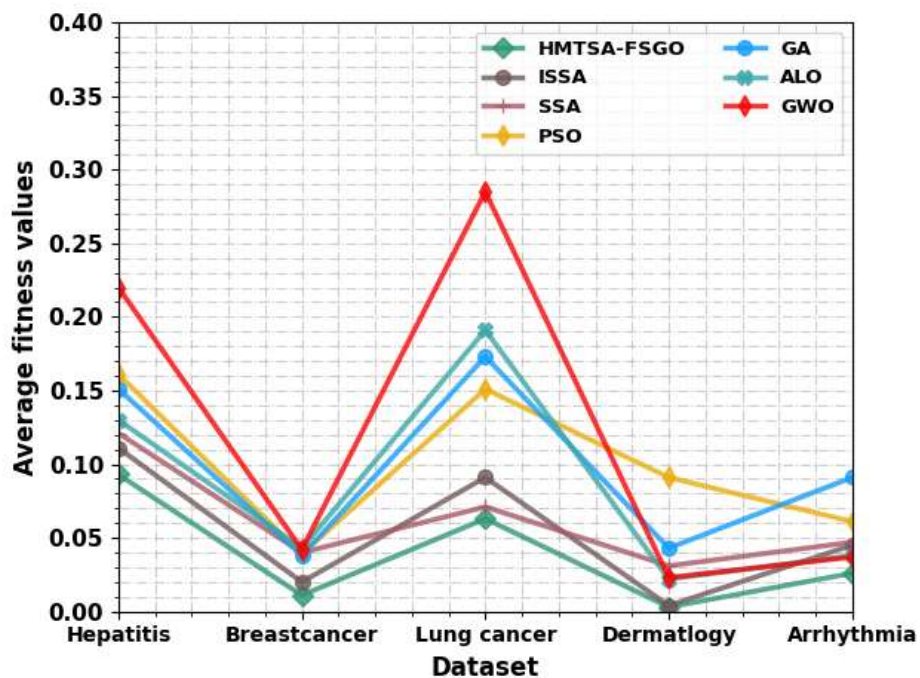
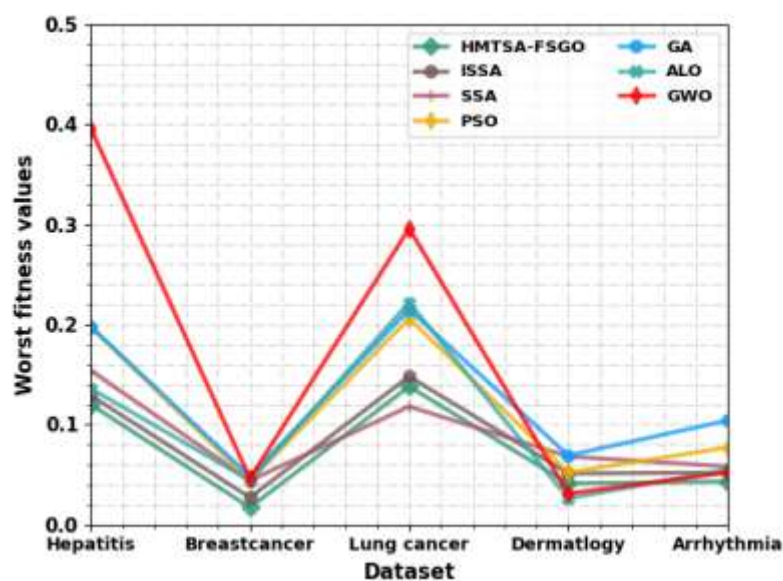
**Figure 5.** AFV analysis of the HMTSA-FSGO technique under five datasets.

Table 5 and Figure 6 represent the worst fitness values (WFVs) of the HMTSA-FSGO technique with other FS methods. The outcomes indicate that the ALO and GWO techniques have shown worse WFV values on all datasets. On continuing with, the SSA, PSO, and GA models have obtained closer WFV values. Even though the ISSA has attained reasonable WFV, the HMTSA-FSGO method illustrates optimum WFV values. It is noticed that the HMTSA-FSGO approach obtains a minimum WFV of 0.072 while the ISSA, SSA, PSO, GA, ALO, and GWO approaches gain greater WFV of 0.081, 0.089, 0.115, 0.127, 0.097, and 0.164, correspondingly.

Table 5. WFV analysis of HMTSA-FSGO technique with other FS models under five datasets.

Dataset	WFVs						
	HMTSA-FSGO	ISSA	SSA	PSO	GA	ALO	GWO
Hepatitis	0.120	0.129	0.154	0.197	0.198	0.136	0.396
Breast cancer	0.017	0.027	0.044	0.044	0.048	0.044	0.046
Lung cancer	0.138	0.148	0.118	0.206	0.215	0.222	0.296
Dermatology	0.041	0.051	0.068	0.052	0.069	0.026	0.031
Arrhythmia	0.043	0.052	0.058	0.077	0.104	0.055	0.052
Average	0.072	0.081	0.089	0.115	0.127	0.097	0.164

**Figure 6.** WFV analysis of the HMTSA-FSGO technique under five datasets.

4.3. Classifier results of HMTSA-FSGO technique

Table 6 and Figure 7 portray the overall classifier results of the HMTSA-FSGO technique with other models. The results highlighted that the HMTSA-FSGO technique gains improved $accu_y$ values. With the hepatitis dataset, the HMTSA-FSGO technique offers a higher $accu_y$ of 93.01% while the ISSA, SSA, PSO, GA, ALO, and GWO models obtain lower $accu_y$ values of 91.30%, 89.44%, 87.55%, 86.10%, 88.32%, and 84.84%, correspondingly. Moreover, the HMTSA-FSGO method offers a maximum $accu_y$ of 97.39% with the breast cancer dataset. The ISSA, SSA, PSO, GA, ALO, and GWO techniques attain minimum $accu_y$ values of 95.76%, 95.56%, 95.56%, 95.17%, 95.07%, and 95.34%, correspondingly. Furthermore, the HMTSA-FSGO method provides a maximum $accu_y$ of 99.15% with the dermatology dataset. The ISSA, SSA, PSO, GA, ALO, and GWO approaches attain minimum $accu_y$ values of 98.31%, 96.29%, 96.51%, 90.76%, 93.28%, and 94.95%, correspondingly. Finally, the HMTSA-FSGO method offers the highest $accu_y$ with the arrhythmia dataset of 67.81%. The ISSA, SSA, PSO, GA, ALO, and GWO techniques attain the lowest $accu_y$ values of 66.05%, 63.85%, 58.08%, 57.11%, 54.68%, and 56.46%, correspondingly.

Table 6. $Accu_y$ analysis of the HMTSA-FSGO technique with other models under five datasets.

Dataset	Accuracy (%)						
	HMTSA-FSGO	ISSA	SSA	PSO	GA	ALO	GWO
Hepatitis	93.01	91.30	89.44	87.55	86.10	88.32	84.84
Breast cancer	97.39	95.76	95.56	95.56	95.17	95.07	95.34
Lung cancer	61.59	59.85	60.30	48.26	56.34	50.62	50.18
Dermatology	99.15	98.31	96.29	96.51	90.76	93.28	94.95
Arrhythmia	67.81	66.05	63.85	58.08	57.11	54.68	56.46

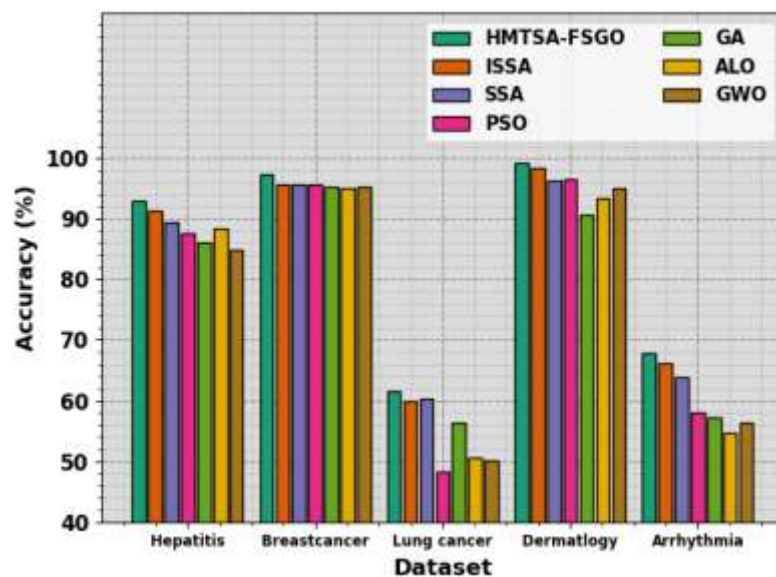


Figure 7. $Accu_y$ analysis of the HMTSA-FSGO technique under five datasets.

The classifier outcome of the HMTSA-FSGO model is graphically shown in Figure 8 for training accuracy (TRAAC) and validation accuracy (VALAC) curves under five datasets. The figure shows valuable insight into the behavior of the HMTSA-FSGO method over different epochs, validating its generalization capabilities and learning process. Notably, the figure indicates a consistent development in the TRAAC and VALAC with increasing epochs. It ensures the adaptive nature of the HMTSA-FSGO approach in the pattern detection technique on both datasets. The increasing tendency in VALAC describes the capability of the HMTSA-FSGO approach to adapt to the TRA data. Also, it offers the correct classification of hidden data, pointing out the strong generalizability.

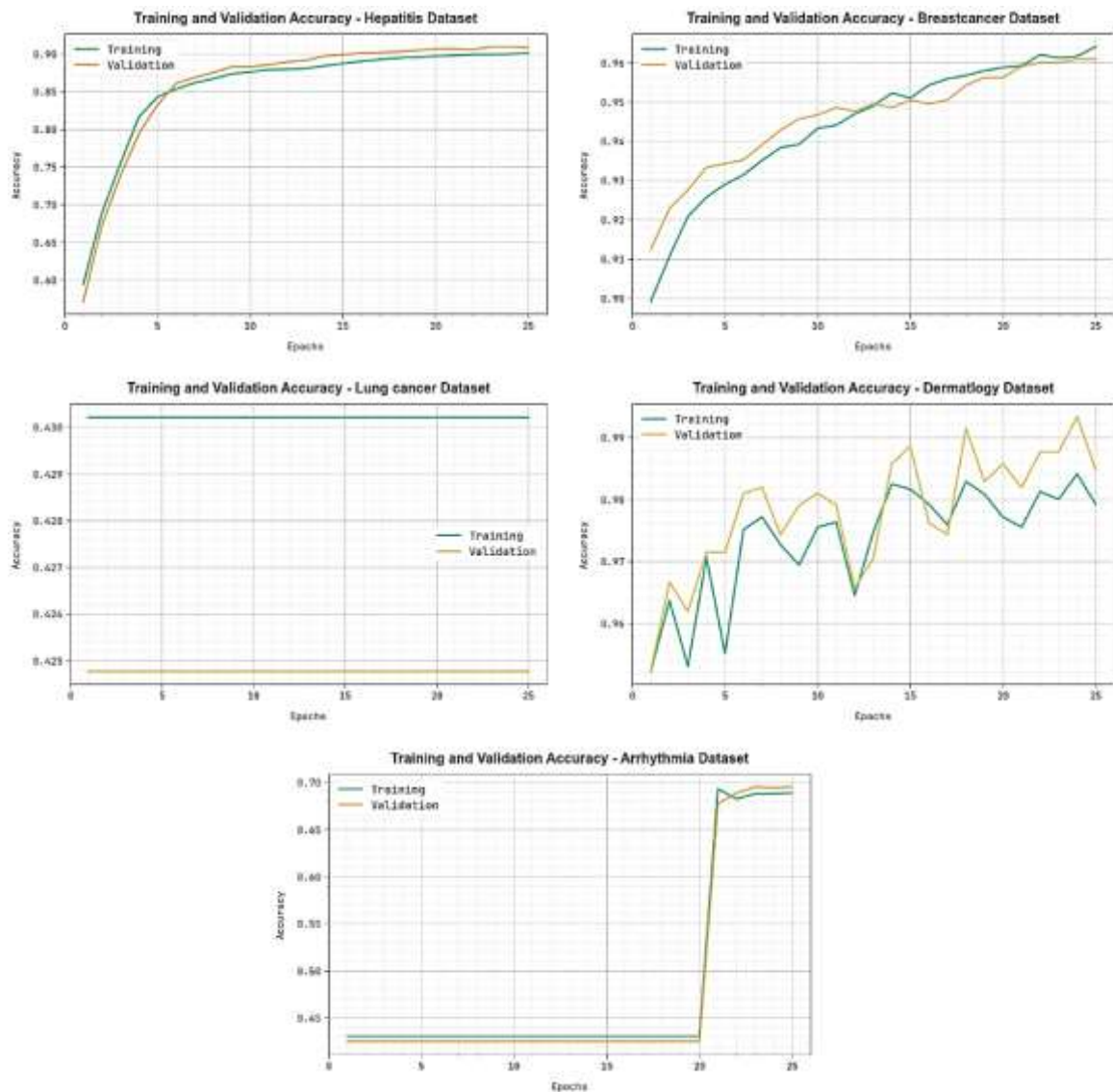


Figure 8. $Accu_y$ curve of HMTSA-FSGO technique under five datasets.

Figure 9 represents the training loss (TRALS) and validation loss (VALLS) outcomes of the HMTSA-FSGO technique over distinct epochs under five datasets. The progressive decline in TRALS highlights the HMTSA-FSGO technique, minimizing the classification error and enhancing the weights on both datasets. The figure clearly shows the HMTSA-FSGO model's relationship with the TRA dataset, highlighting its proficiency in capturing patterns within both datasets. Notably, the HMTSA-FSGO approach constantly progresses its parameters in decreasing the variances amongst the prediction and real TRA classes.

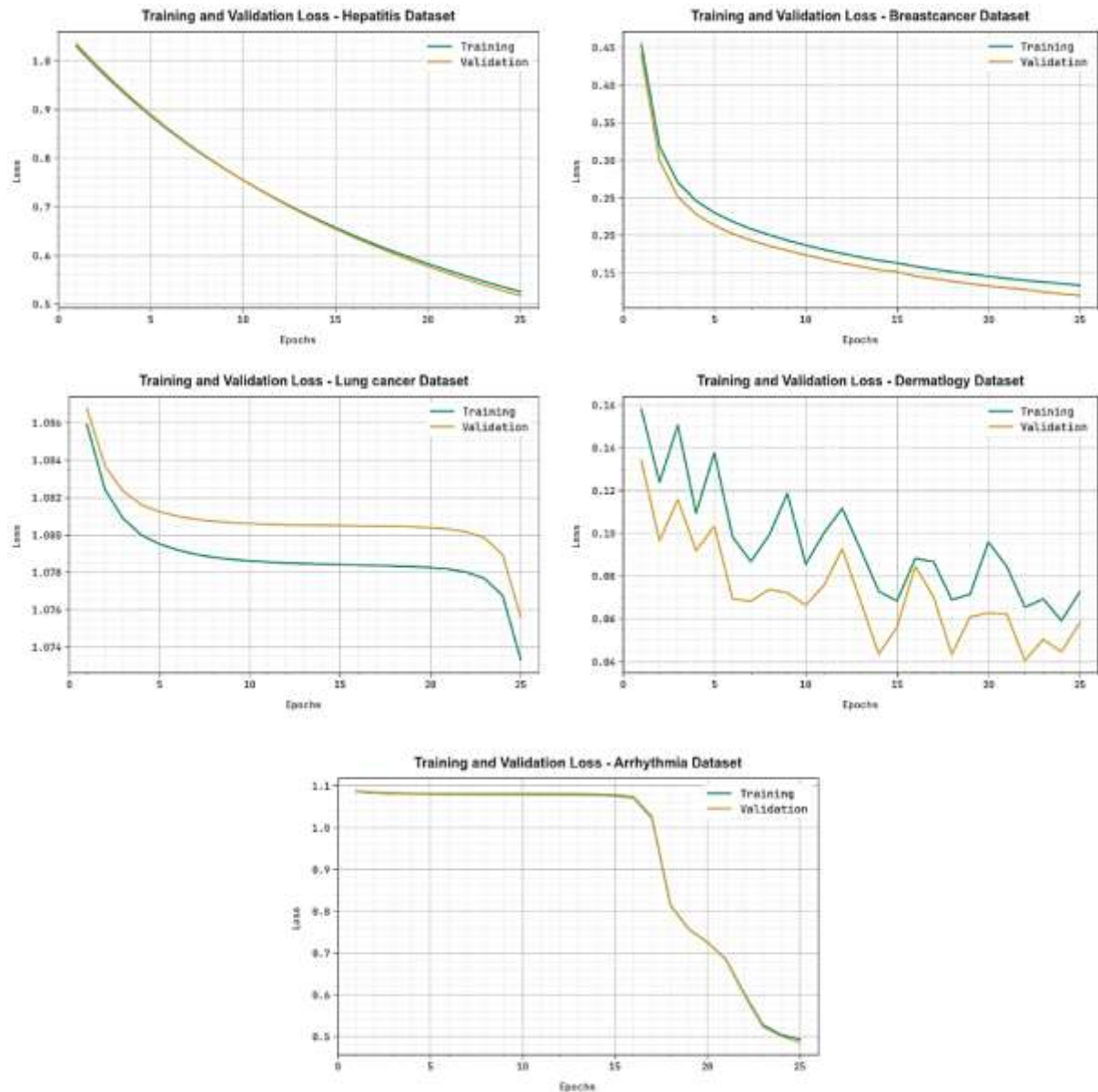


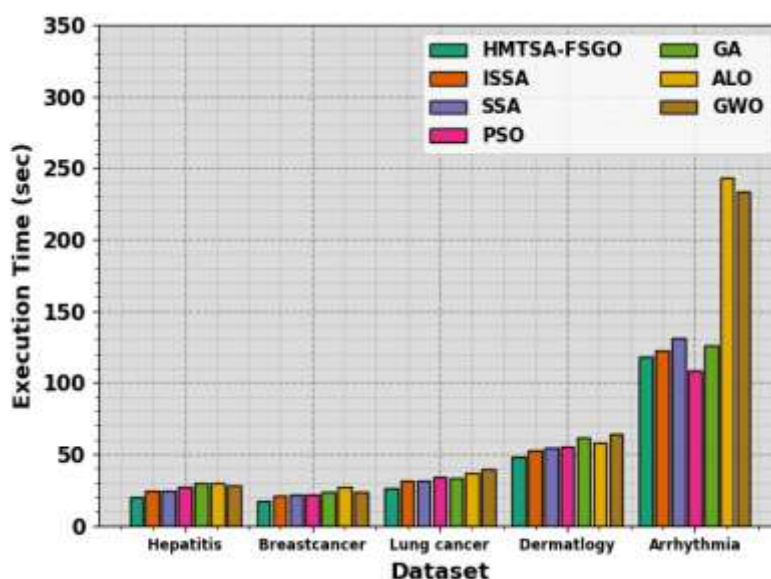
Figure 9. Loss curve of HMTSA-FSGO technique under five datasets.

4.4. Execution time analysis

Finally, the execution time (ET) of the HMTSA-FSGO technique with other models is shown in Table 7 and Figure 10. The experimental values highlighted that the HMTSA-FSGO technique offered the least ET values. With the hepatitis dataset, the HMTSA-FSGO technique provides a lower ET of 20.38s, while the ISSA, SSA, PSO, GA, ALO, and GWO models gain higher ET of 24.04s, 24.22s, 27.26s, 29.42s, 29.42s, and 28.21s, correspondingly.

Table 7. ET analysis of the HMTSA-FSGO technique with other models under five datasets.

Dataset	ET (sec)						
	HMTSA-FSGO	ISSA	SSA	PSO	GA	ALO	GWO
Hepatitis	20.38	24.04	24.22	27.26	29.42	29.41	28.21
Breast cancer	17.37	21.25	21.77	21.76	23.47	27.20	23.84
Lung cancer	26.29	31.84	31.28	33.76	33.55	36.54	39.78
Dermatology	48.44	52.84	54.09	55.26	61.30	57.86	64.54
Arrhythmia	87.82	122.25	130.85	108.67	125.82	242.84	233.23

**Figure 10.** ET analysis of the HMTSA-FSGO technique under five datasets.

Furthermore, the HMTSA-FSGO method provides the lowest ET of 87.82s with the arrhythmia dataset. In contrast, the ISSA, SSA, PSO, GA, ALO, and GWO techniques gain the highest ET of 122.25s, 130.85s, 108.67s, 125.82s, 242.84s, and 233.23s, correspondingly. Therefore, the HMTSA-FSGO technique can be employed to enhance the classification process.

5. Conclusions

In this study, an HMTSA-FSGO technique is presented. The proposed HMTSA-FSGO model mainly aims to eradicate unwanted features and choose the relevant ones that highly impact the classifier results. The HMTSA-FSGO model, the HMTSA is derived by integrating the standard TSA with two concepts: dynamic s-best mutation operator for the optimal trade-off between exploration and exploitation and directional mutation rule for enhanced search space exploration. The HMTSA-FSGO model also includes a BiLSTM classifier to examine the impact of the FS process. RSO can choose the hyperparameters to boost the BiLSTM network performance. The simulation analysis of the HMTSA-FSGO model is tested using a series of experiments. The empirical values highlighted that the HMTSA-FSGO model gains better performance over other models.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The Vice Presidency funded this research/project for Graduate Studies, Business, and Scientific Research (GBR) at Dar Al-Hekma University, Jeddah. The author acknowledges GBR for its technical and financial support.

Data availability statement

The data supporting this study's findings are available at <https://archive.ics.uci.edu/datasets>, reference number [30].

Conflict of interest

The author declares that there is no conflict of interest.

References

1. M. Sharawi, H. M. Zawbaa, E. Emary, H. M. Zawbaa, E. Emary, Feature selection approach based on whale optimization algorithm, in 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), 163–168. <https://doi.org/10.1109/ICACI.2017.7974502>
2. G. I. Sayed, A. Darwish, A. E. Hassanien, A new chaotic whale optimization algorithm for features selection, *J. Classification*, **35** (2018), 300–344. <https://doi.org/10.1007/s00357-018-9261-2>
3. K. Chen, F. Y. Zhou, X. F. Yuan, Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection, *Expert Syst. Appl.*, **128** (2019), 140–156. <https://doi.org/10.1016/j.eswa.2019.03.039>
4. M. Ragab, Hybrid firefly particle swarm optimisation algorithm for feature selection problems, *Expert Syst.*, **41** (2024), e13363. <https://doi.org/10.1111/exsy.13363>
5. H. Faris, M. A. Hassonah, A. M. Al-Zoubi, S. Mirjalili, I. Aljarah, A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture, *Neural Comput. Appl.*, **30** (2018), 2355–2369. <https://doi.org/10.1007/s00521-016-2818-2>
6. S. Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, *Soft Comput.*, **22** (2018), 811–822. <https://doi.org/10.1007/s00521-016-2818-2>
7. Q. Tu, X. Chen, X. Liu, Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection, *IEEE Access*, **7** (2019), 78012–78028. <https://doi.org/10.1109/ACCESS.2019.2921793>
8. F. Hafiz, A. Swain, N. Patel, C. Naik, A two-dimensional (2-D) learning framework for particle swarm based feature selection, *Pattern Recognit.*, **76** (2018), 416–433. <https://doi.org/10.1016/j.patcog.2017.11.027>

9. M. Ragab, Multi-Label scene classification on remote sensing imagery using modified Dingo Optimizer with deep learning, *IEEE Access*, **12** (2024), 11879–11886. <https://doi.org/10.1109/ACCESS.2023.3344773>
10. R. C. T. De Souza, L. D. S. Coelho, C. A. De Macedo, J. Pierezan, A V-Shaped binary crow search algorithm for feature selection, in 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, 1–8. <https://doi.org/10.1109/CEC.2018.8477975>
11. E. H. Houssein, A. Hammad, M. M. Emam, A. A. Ali, An enhanced Coati Optimization Algorithm for global optimization and feature selection in EEG emotion recognition, *Comput. Biol. Med.*, **173** (2024), 108329. <https://doi.org/10.1016/j.compbimed.2024.108329>
12. M. Chaibi, L. Tarik, M. Berrada, A. El Hmadi, Machine learning models based on random forest feature selection and Bayesian optimization for predicting daily global solar radiation, *Inter. J. Renew. Energy D.*, **11** (2022), 309. <https://doi.org/10.14710/ijred.2022.41451>
13. R. R. Mostafa, M. A. Gaheen, M. Abd ElAziz, M. A. Al-Betar, A. A. Ewees, An improved gorilla troops optimizer for global optimization problems and feature selection, *Knowl.-Based Syst.*, **269** (2023), 110462. <https://doi.org/10.1016/j.knosys.2023.110462>
14. B. D. Kwakye, Y. Li, H. H. Mohamed, E. Baidoo, T. Q. Asenso, Particle guided metaheuristic algorithm for global optimization and feature selection problems, *Expert Syst. Appl.*, **248** (2024), 123362. <https://doi.org/10.1016/j.eswa.2024.123362>
15. E. H. Houssein, M. E. Hosney, D. Oliva, E. M. Younis, A. A. Ali, W. M. Mohamed, An efficient discrete rat swarm optimizer for global optimization and feature selection in chemoinformatics, *Knowl.-Based Syst.*, **275** (2023), 110697. <https://doi.org/10.1016/j.knosys.2023.110697>
16. M. Qaraad, S. Amjad, N. K. Hussein, M. A. Elhosseini, 2022. Large scale salp-based grey wolf optimization for feature selection and global optimization, *Neural Comput. Appl.*, **34** (11), 8989–9014. <https://doi.org/10.1007/s00521-022-06921-2>
17. L. Abualigah, M. Altalhi, A novel generalized normal distribution arithmetic optimization algorithm for global optimization and data clustering problems, *J. Amb. Intel. Hum. Comput.*, **15** (2024), 389–417. <https://doi.org/10.1007/s12652-022-03898-7>
18. B. Xu, A. A. Heidari, Z. Cai, H. Chen, Dimensional decision covariance colony predation algorithm: global optimization and high– dimensional feature selection, *Artif. Intell. Rev.*, **56** (2023), 11415–11471. <https://doi.org/10.1007/s10462-023-10412-8>
19. T. Si, P. B. Miranda, U. Nandi, N. D. Jana, S. Mallik, U. Maulik, Opposition-based Chaotic Tunicate Swarm Algorithms for Global Optimization, *IEEE Access*, **12** (2024), 18168–18188. <https://doi.org/10.1109/ACCESS.2024.3359587>
20. G. Liu, Z. Guo, W. Liu, B. Cao, S. Chai, C. Wang, MSHHOTSA: A variant of tunicate swarm algorithm combining multi-strategy mechanism and hybrid Harris optimization, *Plos one*, **18** (2023), e0290117. <https://doi.org/10.1371/journal.pone.0290117>
21. A. Alizadeh, F. S. Gharehchopogh, M. Masdari, A. Jafarian, An improved hybrid salp swarm optimization and African vulture optimization algorithm for global optimization problems and its applications in stock market prediction, *Soft Comput.*, **28** (2024), 5225–5261. <https://doi.org/10.1007/s00500-023-09299-y>
22. Z. Pan, D. Lei, L. Wang, A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling, *IEEE T. Cybernetics*, **52** (2020), 5051–5063. <https://doi.org/10.1109/TCYB.2020.3026571>

23. F. Zhao, S. Di, L. Wang, A hyperheuristic with Q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem, *IEEE T. Cybernetics*, **53** (2022), 3337–3350. <https://doi.org/10.1109/TCYB.2022.3192112>
24. F. Zhao, C. Zhuang, L. Wang, C. Dong, An Iterative Greedy Algorithm With \$ Q \$-Learning Mechanism for the Multiobjective Distributed No-Idle Permutation Flowshop Scheduling, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024. <https://doi.org/10.1109/TSMC.2024.3358383>
25. V. Chandran, P. Mohapatra, A Novel Multi-Strategy Ameliorated Quasi-Oppositional Chaotic Tunicate Swarm Algorithm for Global Optimization and Constrained Engineering Applications, *Heliyon*, 2024. <https://doi.org/10.1016/j.heliyon.2024.e30757>
26. F. A. Hashim, E. H. Houssein, R. R. Mostafa, A. G. Hussien, F. Helmy, An efficient adaptive-mutated Coati optimization algorithm for feature selection and global optimization, *Alexandria Eng. J.*, **85** (2023), 29–48. <https://doi.org/10.1016/j.aej.2023.11.004>
27. A. S. AL-Ghamdi, M. Ragab, Tunicate swarm algorithm with deep convolutional neural network-driven colorectal cancer classification from histopathological imaging data, *Electron. Res. Arch.*, **31** (2023), 2793–2812. <https://doi.org/10.3934/era.2023141>
28. A. Adamu, M. Abdullahi, S. B. Junaidu, I. H. Hassan, An hybrid particle swarm optimization with crow search algorithm for feature selection, *Machine Learn. Appl.*, **6** (2021), 100108. <https://doi.org/10.1016/j.mlwa.2021.100108>
29. A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, M. Abdel-Basset, Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network, *IEEE Access*, **7** (2019), 23319–23328. <https://doi.org/10.1109/ACCESS.2019.2899260>
30. I. M. Batiha, B. Mohamed, Binary rat swarm optimizer algorithm for computing independent domination metric dimension problem, 2024. <https://doi.org/10.1109/ACCESS.2019.2899260>
31. <https://archive.ics.uci.edu/datasets>
32. A. E. Hegazy, M. A. Makhlof, G. S. El-Tawel, Improved salp swarm algorithm for feature selection, *J. King Saud Univ-Com*, **32** (2020), 335–344. <https://doi.org/10.1109/ACCESS.2019.2899260>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)